Towards Fast Image-Based Localization on a City-Scale

Torsten Sattler[1], Bastian Leibe[2], and Leif Kobbelt[1]

[1] RWTH Aachen University, Aachen, Germany,
{tsattler,kobbelt}@cs.rwth-aachen.de
[2] UMIC Research Centre, RWTH Aachen University, Aachen, Germany,
leibe@umic.rwth-aachen.de

Abstract. Recent developments in Structure-from-Motion approaches allow the reconstructions of large parts of urban scenes. The available models can in turn be used for accurate image-based localization via pose estimation from 2D-to-3D correspondences. In this paper, we analyze a recently proposed localization method that achieves state-of-the-art localization performance using a visual vocabulary quantization for efficient 2D-to-3D correspondence search. We show that using only a subset of the original models allows the method to achieve a similar localization performance. While this gain can come at additional computational cost depending on the dataset, the reduced model requires significantly less memory, allowing the method to handle even larger datasets. We study how the size of the subset, as well as the quantization, affect both the search for matches and the time needed by RANSAC for pose estimation.

Keywords: image localization, image retrieval, camera pose estimation

1 Introduction

Image-based localization methods try to estimate the position from which a query image was taken. Once obtained, the position can be used to determine, e.g., the position of a pedestrian [20, 29, 40, 7] or of a mobile robot [11, 12, 37]. An especially interesting application is image-based localization for mobile devices, where a user simply sends a photo taken with her mobile phone to a server and in return receives information about her position [7]. Camera positions computed by localization methods are also useful for Structure-from-Motion reconstructions [1, 10, 14, 18, 28, 35] or for the visualization of photo collections [34].

In order to enable image-based localization, some kind of visual representation of the scene is required. Traditionally, the chosen representation has been a set of images, enabling the use of image retrieval methods to efficiently find similar images and then use the (GPS) positions of the images as an approximation to the position of the query camera. Such a representation usually contains a lot of redundant information as multiple images cover the same part of the scene. Furthermore, many confusing features found in the images have to be removed

This is a preprint for the paper published in F. Dellaert et al. (Eds.): Real-World Scene Analysis 2011, LNCS 7474, pp. 191-211 2012.

for better retrieval [22]. The redundancy in the image set can be exploited to obtain a Structure-from-Motion (SfM) reconstruction of the scene [1, 10, 14, 18, 28, 35], resulting in a 3D point cloud. The image matching part of the SfM pipeline automatically removes most of the confusing features. Thus, a 3D reconstruction offers a more compact representation of the scene than the original images.

While a purely image-based representation only allows to compute the position of the camera, using a 3D model to represent the scene offers the additional advantage that the full camera pose, i.e., both position and orientation, can be determined. Essential for camera pose estimation are correspondences between 2D features in the query image and 3D points in the model. For every 3D point there is a list of 2D image features obtained from the images used to triangulate the point. These features model the appearance of the point from multiple viewpoints under varying lighting conditions. By also extracting local features in the query image, the correspondence search can be modeled as a descriptor matching problem. Due to the large scale of the reconstructions, containing one million or more points, the search method needs to be efficient. It has to find enough correspondences to allow pose estimation. At the the same time, it has to find mainly correct correspondences in order to avoid spending too much time on RANSAC-based pose estimation. A common approach for fast correspondence search is to first find an intermediate representation to quickly narrow down the search for possible correspondence, for example by only considering points found in database images similar to the query image [20].

Recently, Sattler et al. showed that direct search approaches that consider all 3D points with similar enough descriptors as potential correspondences for a feature in the query image achieve a better localization performance, i.e., are able to localize more images [30]. They propose a direct search method based on a visual vocabulary which limits the correspondence search of a query feature to all 3D points with descriptors assigned to the same visual word. Combined with a prioritization scheme, their approach is able to outperform current state-of-the-art methods either in localization performance or efficiency or both. In this paper, we look at two aspects of the method that are critical for scalability to larger datasets: First, the method requires to keep multiple descriptors for every 3D point in memory for efficient nearest neighbor search. Second, as more 3D points are used, the space containing the descriptors becomes denser. As the method uses SIFT features together with SIFT ratio-test [24] to reject wrong correspondences, a denser search space will most likely remove more correct correspondences as well. A simple way to reduce the memory footprint is to use only a subset of the 3D points available in the model. Using fewer points, and thus fewer descriptors, can also have a positive effect on the localization performance for larger datasets if the descriptor space also becomes sparser. In this paper, we experimentally evaluate the impact of considering subsets of the points in the model, selected by a simple reduction scheme recently proposed by Li et al. [23]. More specifically, we explore the relation between the number of points used, localization performance and efficiency, as well as localization accuracy. We show that we can achieve a similar registration performance at

comparable efficiency and slightly better accuracy when using less than half of the points originally contained in the model. To explore the effect of using fewer points on the descriptor space, we simulate a larger dataset by combining multiple smaller ones. Our experiments show that using subsets of the points cannot prevent the descriptor space from becoming too dense, but can speed up the registration process while maintaining a similar registration performance.

We use the notation introduced in [23], referring to 2D local features found in images and their descriptors as *features* and to 3D points and their descriptors from the database images as *points*. A *visual vocabulary* is obtained by clustering a set of local features using approximate k-means [27]. The obtained cluster centers are called *visual words*. Assigning a feature to its visual words means finding the cluster center which has the closest Euclidean distance to the feature through approximate nearest neighbor search.

The paper is structured as follows. Section 2 reviews related work. Section 3 discusses the approaches from [23, 30] in more detail as they are the most relevant work to the work presented in this paper. We experimentally evaluate the combination of the method from [30] and the point filtering proposed in [23] in Section 4. Section 5 concludes the paper by discussing future work.

2 Related Work

Robertson and Cipolla developed one of the earliest image-based methods for localization. Their database consists of 200 image of facades in an urban environment, which are rectified to allow invariance against viewpoint changes [29]. The approach of Zhang & Kosecka retrieves the two images in a database that are most similar to a given query image [40], but instead of canonic views they use SIFT features to handle viewpoint differences. The position of the query camera is then triangulated from the GPS positions of the two retrieved images. Schindler et al. use 30k images, each one associated to a GPS position, to model large parts of a city [31]. To scale their localization method to such a large dataset, they accelerate the image retrieval step through the vocabulary tree method developed by Nister and Stewenius [26], using only features that are informative about their location to obtain a discriminative vocabulary. While Schindler et al. operate on a visual word level, Zamir and Shah use the original SIFT descriptors found in 100k database images, storing the descriptors in a tree-structure [39]. They propose an adapted SIFT ratio-test to deal with repetitive features and achieve positional accuracy comparable to GPS using a voting scheme. To handle an ever larger dataset of around 1 million images, Avrithis et al. aggregate the information of multiple images depicting the same scene into scene maps [4]. This clustering has the positive effect that it increases the recall while reducing the number of documents in the database. A still larger, planet-scale level with more than 6 million database images is considered by Hays and Efros who achieve localization through finding the modes of a probability distribution of possible locations all over the globe [19].

In robotics, the scene in which a robot operates might not be known in advance. In this case, cameras mounted on the robot are used to build a 3D reconstruction of the environment. This model is in turn used to estimate the relative position and orientation of the robot. An early version of such a simultaneous localization and mapping (SLAM) system has been proposed by Se et al. [32]. Current state-of-the-art methods such as [6, 11, 12] try to adapt the SLAM approach to increasingly large scenes for real-time localization.

For large scenes, the construction of the 3D model cannot be achieved in real-time anymore. In case of a static environment the reconstruction can be precomputed using Structure-from-Motion techniques. Irschara et al. propose an approach that uses such models for image-based localization [20]. To narrow down the set of points that have to be considered to establish 2D-to-3D correspondences, they use an image retrieval step to find similar images from the set of images used for the reconstruction. Efficient GPU implementations for both feature matching and vocabulary tree-based retrieval enable their approach to perform in real-time. In order to localize query images substantially different from the database images, Irschara et al. place synthetic cameras on the ground plane to generate additional views. A informative subset of images is picked from the set of original and new images to form the database for retrieval. Wendel et al. generalize the placement of virtual cameras to full 3D to use a similar pipeline for the localization of aerial vehicles [37]. In another retrieval-based approach, Arth et al. use manually selected 3D point sets together with the images the points are visible in for pose estimation on mobile phones [2].

Li et al. show that directly establishing 3D-to-2D correspondences without the intermediate image retrieval step improves localization performance [23]. Starting with points visible in many database images, their prioritized matching algorithm tries to match 3D points to the 2D features in the query image. A point selection schemes computes a more compact representation of the original reconstruction. They show that using such a reduced model improves both localization performance and registration time. Sattler et al. present another approach that directly tries to establish correspondences [30]. In contrast to Li et al. they perform 2D-to-3D matching of 2D features against 3D points. To accelerate the correspondence search they use a prioritization scheme that first evaluates features for which only a small part of the descriptor space has to be searched. The search cost associated with each feature is estimated using a quantization of descriptor space defined by a visual vocabulary.

3 Prioritized Search

In this paper we evaluate the combination of the localization method from Sattler et al. [30] with the point selection scheme from Li et al. [23], aiming to achieve a similar localization performance and efficiency using fewer points and thus less memory. In the following, we review both approaches.

Both methods are based on the key observation that not all 2D-to-3D correspondences that can be found are needed to successfully estimate the camera

pose. The search time can be reduced by applying a prioritization scheme that first considers the most promising features and stops the search if enough correspondences are found. As Li et al. and Sattler et al. perform matching in opposite directions, their prioritization schemes are fundamentally different.

Li et al. try to match 3D points against 2D image features (3D-to-2D matching). They establish a correspondence between a 3D point with mean descriptor d and a 2D feature with descriptor d_1 if the SIFT ratio-test $\frac{\|d-d_1\|_2}{\|d-d_2\|_2} < 0.7$ is fulfilled. Here d_1 and d_2 are the first and second nearest neighbors for d amongst the descriptors in the query image, found through approximate tree-based search [3]. Their prioritization scheme is based on co-visibility of points since a match found for a point p increases the likelihood to find a correspondence for points visible together with p. To this end, two points p and p' are considered to be visible together if there is at least one database image that contains both points. The initial priority of a point is related to the number of database images it is visible in. In case the model was constructed from images obtained from a photo-sharing website, the method thus favors points visible in regions where many photos were taken, i.e., regions which seem to be interesting for tourists. If the model was built from more evenly distributed images, e.g., street view panoramas, stable points visible under different viewing angles are preferred. When a correspondence for p is found the priority of a point p' is increased if p and p' are visible together. The search for correspondences is stopped as soon as $N_t = 100$ correspondences are found. Observing that about one out of every 500 point creates a correspondences by pure chance, Li et al. stop the search as soon as $500 \cdot N_t = 50{,}000$ points have been considered [23].

Large-scale reconstructions contain millions of 3D points and some query images might see only 3D points whose priority is so low that the search would be stopped before any of them are considered. To circumvent this problem, Li et al. propose to use a set of "seed" points [23] that contains locally important points from all over the model. By giving these points a higher priority than all other 3D points, they perform a breadth-first search on the set of seed points to quickly converge to the area of the model that is likely to be seen in the image [23]. The set of seed points is computed by solving a set cover problem, where every point covers all images it is visible in. The seed set is constructed by finding a (minimal) set of points such that every image in the database is covered by at least 5 points. Since computing the minimum set cover is NP-hard, Li et al. use a simple greedy algorithm that iteratively selects the point that covers the largest number of images that have not yet been covered by 5 points [23]. The greedy algorithm is stopped after finding 2000 points to keep the set of seed points compact. Li et al. also use a compact model, again obtained from the greedy algorithm ensuring that every image is covered by at least K points without any limit on the number of selected points, instead of the full 3D model containing all points. They show experimentally that 3-20% of the original features (depending on the structure of the dataset) suffice to achieve both faster localization times and better localization performance, as more images can be registered using the reduced model than with the original model.

While Li et al. match points against features, Sattler et al. propose an approach that performs matching in the other direction (2D-to-3D matching) [30]. They observe that a simple method that stores the mean descriptor for every 3D point in a kd-tree and then performs approximate search [25] for the two nearest neighbors for every query feature, followed by applying the SIFT ratio-test and RANSAC-based pose estimation, achieves better localization performance than current state-of-the-art methods [23]. While offering excellent performance, this method is way too slow for practical applications. Sattler et al. argue that this simple method wastes most of its search time on features that have no correspondence to 3D points in the scene. Instead of treating every feature the same way, they propose a prioritization scheme that firsts evaluates features for which one can quickly decide whether they lead to a correspondence or not. The cost of matching a 2D feature against the reconstruction is related to the number of points that have to be considered. To simultaneously limit the search space and estimate the search cost, Sattler et al. quantize the descriptor space of the used SIFT features into visual words using approx. k-means clustering [27]. In an offline process, the descriptors of the 3D points are assigned to visual words and for each word the list of points that have at least one descriptor assigned to it is stored together with the corresponding feature descriptors. Considering only the points assigned to the same visual word allows to relate the search cost of a query feature to the number of points stored in its word.

Given a new query image and the local features extracted from it, the localization method by Sattler et al. first assigns every feature in the image to its nearest visual word using approximate kd-tree search [25]. The list of (feature,word) pairs is then sorted in increasing number of (point,descriptor) pairs assigned to the words during the offline process. The features in the image are considered in this order. Given the currently considered feature f, the method performs a linear search through all (point,descriptor) pairs stored in the visual word the descriptor d_f of f was assigned to. The search finds the two points p, q ($p \neq q$) whose descriptors d_p, d_q are the nearest neighbors of d_f. Similar to [23], a correspondence between the feature f and the point p is established if the SIFT ratio-test $\frac{\|d_f - d_p\|_2}{\|d_f - d_q\|_2} < 0.7$ is fulfilled. Since the 3D model is obtained from a SfM reconstruction, every point has at least two descriptors assigned to it. Therefore, a point can potentially be assigned to multiple visual words. To avoid establishing multiple correspondences containing the same 3D point, a newly found correspondence (f', p) replaces an existing correspondence (f, p) if $\|d_{f'} - d_p\|_2 < \|d_f - d_p\|_2$ and is rejected otherwise. The search for further correspondences is stopped when N_t correspondences are found. Similar to Li et al., the 6-point DLT algorithm [17] is used to estimate the camera pose inside a RANSAC [13] loop. For robust estimation, a randomized RANSAC variant [9] is used in conjunction with a local optimization scheme [8].

Sattler et al. rigorously explore the design space of this method through experiments on the datasets from [20, 23], showing that their method outperforms other state-of-the-art methods such as [20, 23] in either localization performance or efficiency or even both. They explore different strategies to represent 3D points

by their descriptors, reporting that the following two give the best results: The *all descriptors* (all desc.) strategy represents every 3D point by all of its descriptors. As a result, more than one descriptor of a point can be stored in the same visual word, increasing the search time for the word. The *integer mean per visual word* (int. mean) strategy tries to reduce the memory requirements by replacing multiple descriptors of the same point assigned to the same word by their mean descriptor. The entries of this mean descriptor are then rounded to the nearest integer value to be able to use only 1 byte for each entry instead of the 4 bytes needed by a floating point representation. Choosing $N_t = 100$ helps to reduce the search times without any significant negative impact on the registration performance. Furthermore, assuming an initial inlier ratio of 20% for RANSAC effectively limits the maximal number of taken samples with little impact on the localization performance. Source code for the method has been made publicly available and can be found at `http://www.graphics.rwth-aachen.de/localization/`.

There is an interesting analogy between the prioritization scheme of Sattler et al. and the well-known idf-weighting scheme used in image retrieval [33]. The idf-scheme weights down words that are used in many documents since they are less discriminative. Similarly, the prioritization scheme from [30] favors features mapped to a visual word which does not occur very often in the model and thus contains discriminative points. Therefore, besides trying to minimize the search costs by finding a suitable ordering of features, the prioritization scheme starts with the most promising features found in the image.

An interesting result from [30] is that the performance of a generic set of 100k visual words obtained from an unrelated dataset is similar to the performance of a specialized vocabulary trained from the descriptors of the points in the corresponding reconstruction. This means that the same vocabulary can be re-used, independently of the considered dataset. The main cause for this, somewhat surprising, result is that Sattler et al. perform a very approximate nearest neighbor search to compute the assignment of descriptors to visual words in order to minimize search costs. Specially trained vocabularies do not offer any advantages for such a very approximative search.

Two problems will arise when applying the method from Sattler et al. on even larger datasets. Since multiple SIFT descriptors are stored for every 3D point, the model will eventually become too large to fit into the RAM of a PC. As more and more points are used, the distances between the descriptors of one point and their nearest descriptors belonging to another point decrease. This has a positive impact on the run-time of the RANSAC-based pose estimation, because the SIFT ratio-test is able to remove more and more wrong correspondences. However, as the descriptor space becomes denser, the ratio-test will also filter out more correct correspondences. Thus only features with descriptors very similar to the ones of its corresponding 3D point will pass the SIFT ratio-test. As a result, images differing too much from the views in the database cannot be registered anymore, decreasing the localization performance of the algorithm. Compact models containing fewer points than the original reconstruction require less memory and can therefore help to solve the first problem. Using fewer points

Table 1: Details on the datasets used for experimental evaluation.

Dataset	# Cameras	# 3D Points	# Descriptors	# Query Images
Dubrovnik	6044	1,886,884	9,606,317	800
Rome	15,179	4,067,119	21,515,110	1000
Vienna	1324	1,123,028	4,854,056	266

can also induce a sparser descriptor space, helping the localization method to avoid rejecting too many good correspondences.

In the case of 3D-to-2D matching, the descriptor space formed by the 2D features in an image is much sparser than the descriptor space of the 3D model. Thus, Li et al. are able to avoid the problem of rejecting too many correct matches at the cost of finding more wrong correspondences. Note that their approach still has problems scaling to larger datasets. To enable the breadth-first search performed by the algorithm, a larger set of seed points has to be used for reconstructions containing more points. Based to the observation that roughly one out of every 500 points matches by chance, it will happen that the algorithm stops before even considering the whole seed set since enough correspondences are already found. In turn, finding enough good candidate points for matching is not a problem for the method from Sattler et al. due to using a visual vocabulary for finding possible correspondences.

4 Compact Models for 2D-to-3D Search

In this section, we evaluate the combination of the localization method from [30] and the point selection scheme proposed by [23]. Specifically, we explore the impact of compact models constructed by different choices for the set cover parameter K on localization performance, efficiency and accuracy. In Section 4.1 the used datasets and the experimental setup are explained. The impact of the parameter K on both registration performance and registration times is evaluated in Section 4.2. In Section 4.3 we show that compact models can help the method to handle larger datasets. In Section 4.4 we detail the impact of K on the localization accuracy. Since the approach from Sattler et al. outperforms the other state-of-the-art approaches, such as [20, 23], we do not compare our results against other approaches.

4.1 Experimental Setup

We use the three large-scale datasets from [20, 23, 30] to allow a direct and fair comparison. For two of the datasets, Dubrovnik and Rome, the database images for the reconstruction were obtained from the photo-sharing website Flickr [23]. For the Vienna dataset the database images were taken at regular intervals with a single camera [20]. The original Dubrovnik reconstruction consists of 6844 images depicting parts of the old city of Dubrovnik. 800 randomly selected images were removed from the reconstruction to obtain a set of relevant query images.

Table 2: The percentage of points selected depending on K for Dubrovnik and Rome.

	K									
Dataset	100	200	300	400	500	600	700	800	900	1000
Dubrovnik	3.84%	8.61%	13.58%	18.6%	23.55%	28.24%	32.68%	36.88%	40.82%	44.59%
Rome	3.56%	8.36%	13.57%	18.93%	24.23%	29.42%	34.32%	38.94%	43.29%	47.40%

Table 3: The percentage of points selected depending on K for the Vienna dataset.

	K							
Dataset	500	750	1000	1250	1500	2000	2500	3000
Vienna	7.54%	12.53%	18.03%	23.62%	29.20%	39.68%	49.28%	58.00%

For every camera in the test set, the SIFT descriptors of the points visible in it were deleted from the model. Any point visible in only one remaining camera was also removed. The query images for the Rome dataset were obtained in the same fashion, removing 1000 randomly selected images from the 16,179 images in the initial reconstruction. In contrast to the Dubrovnik model the Rome reconstruction consists of multiple connected components, each one representing a distinct landmark in Rome [23]. The Vienna model consists of 1324 cameras in three connected components. Query images were obtained from the Panoramio website. All query images have maximal width and height of 1600 pixels. The Dubrovnik and Rome models used in [23] and [30] differ slightly in the number of 3D points they contain. We use the latter model. More information about the datasets than presented in Table 1 is available in [20, 23].

For the Dubrovnik model, Li et al. computed a transformation into a geo-referenced coordinate frame such that distances in the model can be expressed in meters [23]. Since the query images were obtained by removing images from the initial reconstruction, we can use the original camera positions computed by SfM as ground truth and measure the localization accuracy.

As proposed by Li et al., we accept a query image as localized, or registered against the model, if the best camera pose estimated by RANSAC has at least 12 inliers. Repeating each experiment 10 times to account for the random nature of RANSAC, we report the average number of images that can be registered and the average time needed to register or reject an image. Assuming that SIFT features are already given, the time needed to process an image is the sum of the time needed to assign all of its features to visual words, the time needed for correspondence search and the time needed by RANSAC to estimate the camera pose. Beside the total time, we also report the time required for the correspondence search and the time needed for RANSAC.

4.2 Compact Models

As shown in [38], pictures found on photo collection websites are distributed around certain iconic views as tourists tend to take slightly different photos

Table 4: Mean registration performance and times for **100k** visual words and different values for K. $K=\infty$ denotes the results reported in [30] for which all available points were used. We obtain a similar registration performance as [30] using compact models with of fewer 3D points. For Dubrovnik and Rome we achieve better registration times.

		all descriptors			integer mean per vw		
	K	# reg. images	registr. time [s]	rejection time [s]	# reg. images	registr. time [s]	rejection time [s]
Dubrovnik	100	569.40 ± 3.17	1.79	5.66	604.10 ± 4.61	1.59	5.45
	200	736.20 ± 3.26	0.94	5.01	739.60 ± 1.96	0.78	4.67
	400	776.80 ± 1.75	0.42	3.43	775.30 ± 1.16	0.37	3.03
	600	781.30 ± 1.42	0.31	3.01	778.50 ± 1.18	0.28	2.66
	800	782.10 ± 1.20	0.29	2.32	779.20 ± 1.40	0.26	2.17
	900	782.00 ± 0.94	0.27	2.45	780.80 ± 1.23	0.26	1.96
	1000	781.90 ± 0.99	0.27	2.43	781.30 ± 0.95	0.25	1.88
	∞	783.90 ± 1.60	0.31	2.22	782.00 ± 0.82	0.28	1.70
Rome	100	950.10 ± 1.66	0.41	3.08	947.40 ± 2.76	0.32	2.46
	200	965.20 ± 1.62	0.23	1.84	964.10 ± 1.45	0.20	1.63
	400	971.90 ± 1.45	0.21	1.90	972.50 ± 1.08	0.18	1.77
	600	974.60 ± 1.07	0.21	1.88	974.70 ± 1.83	0.18	1.77
	800	973.90 ± 1.52	0.21	1.76	974.30 ± 1.16	0.18	1.60
	900	974.00 ± 1.33	0.22	1.62	975.90 ± 1.91	0.19	1.56
	1000	974.90 ± 0.99	0.23	1.63	974.80 ± 1.87	0.20	1.59
	∞	976.90 ± 1.29	0.29	1.90	974.60 ± 1.65	0.25	1.66
Vienna	500	122.50 ± 2.07	2.44	5.37	127.00 ± 1.76	2.28	5.12
	1000	181.30 ± 2.00	1.34	4.25	184.70 ± 2.54	1.24	4.02
	1500	194.70 ± 0.82	0.73	3.63	193.90 ± 1.29	0.64	3.50
	2000	202.30 ± 1.34	0.62	3.30	202.00 ± 1.05	0.63	3.04
	2500	206.40 ± 1.26	0.60	3.07	205.10 ± 1.10	0.58	2.85
	3000	206.90 ± 0.74	0.54	2.84	206.10 ± 1.10	0.51	2.70
	∞	207.70 ± 1.06	0.50	2.40	206.90 ± 0.88	0.46	2.43

of the same buildings. Since the query images for Dubrovnik and Rome were obtained by randomly selecting images from the reconstruction, they have the same distribution as the database images. Thus the descriptors found in the query images should be rather similar to those in the model. While the Vienna model was reconstructed from images taken in nearly regular intervals, the query images obtained from Panoramio follow a different distribution. Furthermore, the database images were taken with a single camera on the same day while query images are taken at different days and at different times of day with different cameras. This makes the Vienna dataset the most challenging of the three datasets and we can expect a larger difference between the SIFT descriptors found in the query image and the those found in the database images. Due to this difference in distributions, we use a different range of values for the set cover parameter K for the Vienna dataset compared to the Dubrovnik and Rome datasets, similar to [23]. Table 2 shows the percentage of points selected depending on K for the Dubrovnik and Rome datasets, while Table 3 shows the

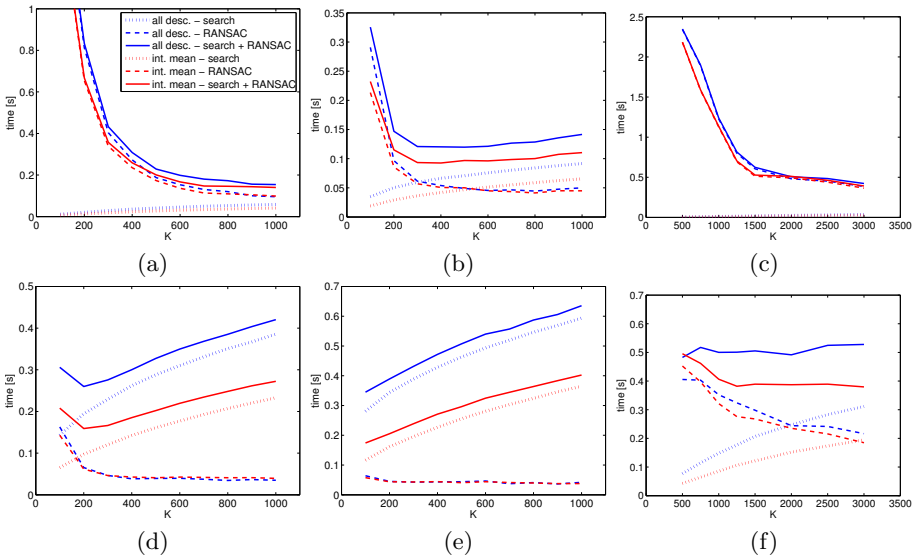

Fig. 1: Dependence of the average time needed to find enough correspondences and the average time to compute the camera pose through RANSAC on the parameter K. Timings are shown for (a), (d) Dubrovnik, (b), (e) Rome and (c), (f) Vienna. 100k visual words were used for the results shown in the top row and 10k words for the bottom row. Fewer 3D points yield more wrong correspondences, increasing the run-time of RANSAC. Search time increases with the number of points in the words.

percentage of selected points for the Vienna dataset. We only consider values for K until obtaining around 50% of the points contained in the original model since we want to use the compact models to save storage space.

We evaluate the compact models obtained for the values for K shown in Tables 2 and 3 together with the two strategies, *all desc.* and *int. mean*, in the pipeline proposed in [30]. The visual vocabulary containing 100k words employed in this experiments is the same as in [30]. We report the mean number of images that can be localized and the mean time needed to register or reject an image in Table 4. Small values for K lead to a significantly worse localization performance with high registration and rejection times. Using more points allows to achieve a registration performance similar to [30]. Slightly faster registration times can be achieved for K from $\{800, 900, 1000\}$ for Dubrovnik and Rome, while the registration times for the Vienna dataset are a little worse compared to the full model. There are two possible explanations for the observed behavior. First, using fewer points and thus fewer descriptors leads to a sparser descriptor space and visual words that are less full. As the distances between descriptors stored in a visual word grow, it becomes more likely to accept wrong matches through the SIFT ratio-test, which in turn increase the registration time. Secondly, the selected points might not suffice to allow robust localization.

Table 5: Mean registration performance and times for **10k** visual words and different values for K. For the Dubrovnik and Rome datasets, fewer points still allow a similar registration performance compared to [30] at higher localization costs. A significantly better performance at comparable registration times is achieved for the Vienna dataset.

		all descriptors			integer mean per vw		
	K	# reg. images	registr. time [s]	rejection time [s]	# reg. images	registr. time [s]	rejection time [s]
Dubrovnik	100	771.00 ± 1.49	0.36	2.03	765.40 ± 1.96	0.26	1.55
	200	778.80 ± 1.75	0.31	1.81	777.20 ± 0.92	0.21	1.18
	300	780.70 ± 0.95	0.33	1.70	778.00 ± 1.49	0.22	1.18
	400	782.60 ± 1.84	0.35	1.86	779.20 ± 1.03	0.24	1.19
	600	783.30 ± 0.95	0.40	2.02	781.60 ± 2.22	0.27	1.38
	800	783.40 ± 0.97	0.44	2.12	783.20 ± 1.62	0.30	1.43
	1000	784.50 ± 1.65	0.47	2.22	784.30 ± 0.82	0.32	1.69
	[30]	783.90 ± 1.60	0.31	2.22	782.00 ± 0.82	0.28	1.70
Rome	100	964.20 ± 1.32	0.38	1.54	959.00 ± 1.56	0.21	1.13
	200	972.40 ± 1.84	0.43	2.01	968.90 ± 1.10	0.24	1.36
	300	974.90 ± 0.99	0.47	2.38	971.60 ± 1.17	0.28	1.44
	400	978.80 ± 1.03	0.51	2.64	974.80 ± 1.55	0.31	1.55
	600	980.00 ± 0.67	0.58	2.76	976.60 ± 0.84	0.36	1.86
	800	978.80 ± 1.03	0.63	3.19	976.80 ± 1.62	0.40	2.09
	1000	980.20 ± 1.75	0.67	3.45	977.10 ± 1.73	0.44	2.23
	[30]	976.90 ± 1.29	0.29	1.90	974.60 ± 1.65	0.25	1.66
Vienna	500	198.70 ± 1.16	0.54	2.63	198.10 ± 1.66	0.55	2.18
	750	209.00 ± 1.25	0.57	2.31	205.90 ± 0.74	0.52	1.86
	1000	215.00 ± 1.25	0.55	1.91	209.60 ± 1.43	0.46	1.61
	1250	217.50 ± 0.97	0.56	1.67	213.60 ± 0.70	0.44	1.37
	1500	218.10 ± 0.88	0.56	1.54	213.90 ± 1.10	0.44	1.35
	2000	219.30 ± 0.67	0.55	1.43	214.80 ± 1.23	0.44	1.27
	2500	219.70 ± 0.82	0.58	1.31	215.20 ± 0.92	0.44	1.23
	3000	218.90 ± 1.20	0.58	1.29	214.10 ± 1.29	0.43	1.23
	[30]	207.70 ± 1.06	0.50	2.40	206.90 ± 0.88	0.46	2.43

The first explanation is easy to verify. Figure 1 shows how the mean time for correspondence search and the mean time RANSAC needs depend on the choice of K for (a) Dubrovnik, (b) Rome and (c) Vienna. Compact models with more points indeed lead to faster RANSAC times due to fewer wrong matches.

To reject the second explanation, we repeat the experiment using a visual vocabulary containing only 10k words. Since each of the words in this smaller vocabulary covers a larger part of descriptor space, the likelihood of assigning the descriptors of a 3D point and the descriptor of its corresponding feature to the same word increases compared to the original vocabulary. Table 5 shows that a good registration performance can be achieved for much lower values of K with the smaller vocabulary, indicating that enough points for robust localization are selected. A compact model containing only about 18% of the original points ($K = 400$ for Dubrovnik and Rome, $K = 1000$ for Vienna) gives a performance

comparable to the original methods from [30], albeit at increased registration times. As shown in Figure 1(d)-(e) this increase is mainly due to the slower search as more points are contained in the words. It is noticeable that the difference in search time for both strategies is much larger for 10k words than for 100k words. Since more descriptors of the same point are mapped to the same word for the smaller vocabulary, the *integer mean* strategy is able to compress them into one mean descriptor while the *all descriptor* strategy has to use all descriptors. At the same time, the *all descriptor* strategy is able to handle denser visual words much better as all information about the 3D points is preserved, which is visible in the better registration performance for smaller values for K. We observe a significant increase in the localization performance for the smaller vocabulary on the Vienna dataset. As mentioned above, the difference in viewpoint and viewing condition is the largest on this dataset, explaining that using fewer words increases the chance of assigning features and points that belong together to the same visual word. As predicted above, the number of wrong correspondences decreases for the words in the smaller vocabulary as evident by the faster RANSAC run-time shown in Figure 1(d)-(f) compared to Figure 1(a)-(c). This faster pose estimation has the largest impact on the Vienna dataset for which the mean registration time was dominated by the time spend by RANSAC when using 100k words.

4.3 Combining the Datasets

As shown in the previous experiments, we can use compact representations of the 3D models obtained by the point selection scheme from [23] to reduce the memory footprint and still obtain a similar registration performance and efficiency compared to the original models. For larger datasets, the descriptor space becomes denser as more points are used. As a result, the SIFT ratio-test is more likely to also reject good correspondences. As compact models contain fewer points, they could help to avoid the loss in registration performance.

In this section we want to explore the effect of using compact models on the density of the descriptor space for datasets larger than the three models used so far. Although modern SfM approaches can efficiently handle large datasets, obtaining the images for very large scenes is still challenging. We therefore try to simulate a larger dataset by combining the three models. This is motivated by the observation that only few correspondences are found when using the query images from one dataset for different model [30]. The combined datasets therefore represents a sort of "best case" model which consists of distinct landmarks. If we can observe that the descriptor space becomes too dense for this model, we would expect that the space will also become too dense for other large datasets.

We combine (subsets of) the three models to obtain three larger datasets: The first one consists of all points from all three datasets, i.e., we set $K = \infty$. The second is obtained using the point selection scheme with $K = 900$ on Dubrovnik and Rome and $K = 2500$ on Vienna. The last one consists of the points selected with $K = 400$ on Dubrovnik and Rome and $K = 1000$ on Vienna. We chose the combinations 900 / 2500 and 400 / 1000 because these were the smallest values for K that gave results similar to the original method when using 100k

Table 6: Results for combining different versions of the three models for query images from each dataset. We use K_1 to build compact models for Dubrovnik and Rome and K_2 to obtain a compact model for Vienna. For comparison we include the results from [30] on the single models. Due to the denser descriptor space, registration performance drops compared to [30] for the combined models, but the usage of compact models can help to decrease the registration and rejection times at a similar localization performance.

			# reg.	registered			rejection
	K_1 / K_2	method	images	search [s]	RANSAC [s]	total [s]	time [s]
Dubrovnik	∞ / ∞	all desc.	779.20 ± 0.63	0.42	0.02	0.55	1.32
		int. mean	776.00 ± 1.25	0.33	0.02	0.46	1.05
	900 / 2500	all desc.	775.80 ± 1.23	0.27	0.02	0.40	0.90
		int. mean	775.80 ± 1.40	0.20	0.02	0.33	0.68
	400 / 1000	all desc.	774.60 ± 1.17	0.19	0.02	0.31	0.64
		int. mean	773.50 ± 0.85	0.13	0.02	0.25	0.45
	[30]	all desc.	783.90 ± 1.60	0.10	0.08	0.31	2.22
		int. mean	782.00 ± 0.82	0.08	0.08	0.28	1.70
Rome	∞ / ∞	all desc.	973.10 ± 2.02	0.24	0.04	0.36	1.68
		int. mean	971.20 ± 1.55	0.19	0.04	0.31	1.35
	900 / 2500	all desc.	975.00 ± 1.25	0.16	0.04	0.28	1.32
		int. mean	970.20 ± 1.23	0.12	0.04	0.24	1.10
	400 / 1000	all desc.	971.90 ± 0.74	0.11	0.04	0.23	1.26
		int. mean	970.90 ± 1.79	0.07	0.04	0.20	1.09
	[30]	all desc.	976.90 ± 1.29	0.15	0.05	0.29	1.90
		int. mean	974.60 ± 1.65	0.11	0.05	0.25	1.66
Vienna	∞ / ∞	all desc.	202.70 ± 0.67	0.54	0.01	0.67	1.29
		int. mean	200.80 ± 0.79	0.43	0.01	0.57	0.98
	900 / 2500	all desc.	203.90 ± 0.74	0.36	0.02	0.50	0.82
		int. mean	200.60 ± 0.52	0.27	0.03	0.41	0.60
	400 / 1000	all desc.	192.60 ± 1.26	0.24	0.02	0.37	0.66
		int. mean	189.10 ± 0.57	0.16	0.02	0.30	0.48
	[30]	all desc.	207.70 ± 1.06	0.06	0.30	0.50	2.40
		int. mean	206.90 ± 0.88	0.05	0.28	0.46	2.43

respectively 10k visual words. We only consider the vocabulary of size 100k words since the search time for 10k words were already too large for the single models. Table 6 reports the registration performance and efficiency for the query images from each dataset and compares it to the results obtained in [30] on the single models. As can be seen, the sparser descriptor space obtained from the compact models is still too dense to prevent a loss in registration performance. However, the compact models can be used to speed up the search times while still obtaining very similar performance compared to using the full model.

The denser descriptor space has a significant impact on the pose estimation time as most wrong correspondences are eliminated by the SIFT ratio-test, allowing us to achieve even better registration and rejection times than the original method. For example, we obtain significantly better registration times with $K = 1000$ for the query images from Vienna dataset when combining the mod-

els compared to only considering the Vienna model. This is again caused by the denser descriptor space which helps to eliminate wrong correspondences.

4.4 Localization Accuracy

We measure the localization accuracy of the combination of point selection and the localization method from [30] on the Dubrovnik dataset. The random nature of RANSAC results in slightly differing camera pose estimates for all repetitions of the experiment. To compensate for this, we measure the average camera position for every query image from all its estimated poses with at least 12 inliers from the 10 repetitions. We report the distance between this averaged position and the ground truth position of the query camera in the original reconstruction.

The greedy point selection algorithm iteratively picks the point that covers the largest number of cameras that have not yet been covered and thus prefers points visible in many cameras [23]. We can expect that the amount of positional uncertainty related to the selected points is relative small, since they have been detected in multiple images. Using these high-quality points should improve the localization accuracy. Unfortunately, SIFT features are not equally distributed over images but mostly found in highly textured regions. Given such a highly textured region, it is rather likely that multiple points in this region appear in many database images. Thus if one of them is selected by the greedy algorithm it is very likely that also the other points are selected since they are also visible in a similar number of images. As a result, it might happen that the selected points are not well-distributed over the model but form small clusters. This in turn can lead to unstable or even degenerate configurations for the pose estimation step. To verify whether using fewer points yields less accurate localization results, we look at the cumulative distribution of the query images over localization errors depicted in Figure 2. In contrast to [30], we followed RANSAC-based pose estimation with a linear least-squares estimate of the pose from the inliers. As seen in Section 4.2, the number of images that can be registered differs with the choice of K. To allow a fair comparison, we normalized the cumulative histogram for each variant using the total number of images that it could register, i.e., the number of images that could be localized at least once during the 10 repetitions of the experiment. As can be seen in the figure, using too few points indeed results in worse localization accuracy. However, about 14% of the total features ($K = 300$, cf. Table 2) are already sufficient to achieve localization accuracy comparable to or better than the results reported in [30]. Choosing K from $\{800, 900, 1000\}$ gives the best results. We notice that using the smaller vocabulary of 10k words improves the accuracy. Due to the coarser quantization and the approximative nature of visual word assignments, it is more likely to assign two descriptors of the same 3D point to the same visual word when using 10k words instead of 100k. This enables the algorithm to find more correspondences for points seen from rather large viewpoint changes compared to the original cameras, which in turn yield better configurations for pose estimation.

More details on selected values for K are given in Table 7. We report the median localization error, the 1st and 3rd quartile and the number of images with

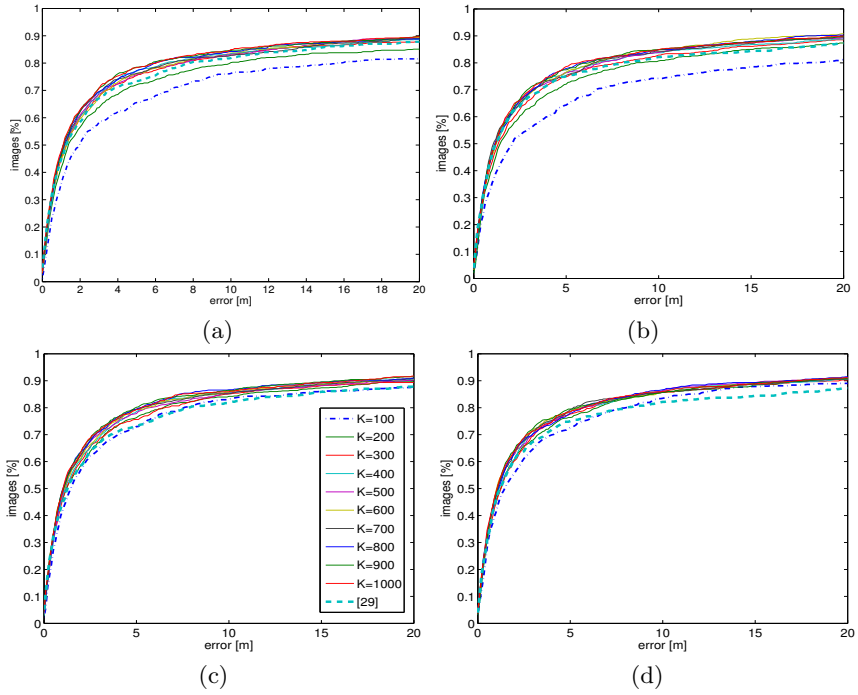

Fig. 2: Normalized cumulative histograms of the distribution of the localization error depending on K for *all descriptors* using (a) 100k words respectively (c) 10k words and *integer mean* using (b) 100k words and (d) 10k words. Choosing $K \geq 300$ helps to improve the localization accuracy compared to the original method independently of the vocabulary size since a higher percentage of reliably localized images points is used. Values for K from the range $[800, 1000]$ give the best results.

a localization error smaller than 18.3m respectively 400m, which correspond to the mean and maximal errors reported in [23]. The results verify the observations from Figure 2, since compact models help to improve the localization accuracy. Again, the usage of a smaller vocabulary has a positive impact on the accuracy of the position estimates. We do not report the mean or maximal registration error, since there are a few images with very high localization error of up to multiple kilometers. These large errors are caused by degenerate point configurations for pose estimation. We observe that images with such large errors mostly have more than 12 inliers, indicating that the pure inlier count is not a good measure for localization accuracy. This behavior has already been reported by Sattler et al. [30]. They show that using the focal length of an image, obtained from its EXIF tag, for 3-point pose estimation [13, 16] or a more restrictive camera model, which estimates only its focal length and a radial distortion parameter [21], help to obtain more accurate estimates. We could also estimate the covariance of the position parameters of the query camera and reject a camera for which the positional uncertainty is too high.

Table 7: Statistics on the localization errors for selected values of K. Using compact models helps to improve the localization accuracy compared to the original methods using all points ($K = \infty$) from [30] and the method from [23].

K	Method	# vw	Median [m]	Quartiles [m] 1st	Quartiles [m] 3rd	#imgs. with error < 18.3m	#imgs. with error > 400m
400	all desc.	10k	1.2	0.5	4.1	710	7
		100k	1.3	0.5	4.3	690	9
	int. mean	10k	1.2	0.5	4.1	703	6
		100k	1.3	0.5	4.5	689	12
800	all desc.	10k	1.1	0.4	3.8	710	9
		100k	1.2	0.5	4.3	698	11
	int. mean	10k	1.2	0.4	4.3	714	12
		100k	1.3	0.4	4.1	705	13
900	all desc.	10k	1.1	0.4	3.6	713	8
		100k	1.2	0.4	3.9	698	10
	int. mean	10k	1.2	0.5	3.5	709	9
		100k	1.3	0.5	4.3	696	14
1000	all desc.	10k	1.1	0.4	3.8	714	9
		100k	1.2	0.4	4.0	700	11
	int. mean	10k	1.1	0.4	4.1	711	11
		100k	1.3	0.5	4.3	701	10
∞	all desc.	100k	1.4	0.4	5.9	685	16
	int. mean	100k	1.3	0.5	5.1	675	13
100	P2F [23]	-	9.3	7.5	13.4	655	-

We report the localization accuracies for the combined datasets in Table 8. The results were obtained without the final linear least-square pose estimate and show no significant difference in localization accuracy between the different combinations and the original results from [30], obtained using only the Dubrovnik dataset. The drop in localization accuracy compared to the experiments on the Dubrovnik dataset alone (cf. Table 7) can be explained by the different set of correspondences found when also using the points from the other datasets.

5 Conclusion & Future Work

In this paper we have shown that not all points contained in a Structure-from-Motion model are needed for robust image-based localization. By combining the state-of-the-art localization method from Sattler et al. [30] and the simple point selection scheme from Li et al. [23] we demonstrated that using less than half of the original points still allows state-of-the-art localization performance at similar registration and rejection times and with slightly better localization accuracy. This result is still valid when combining the different datasets to simulate one larger reconstruction. Therefore, we can save memory by storing fewer points and descriptors without a significant sacrifice in performance and efficiency. As the method of computing the compact models does not depend on the type

Table 8: Statistics on the localization errors for the combined datasets from Section 4.3. There is no significant difference in localization accuracy between the different combinations and the original results from [30].

K_1 / K_2	Method	Median [m]	Quartiles [m] 1st	Quartiles [m] 3rd	#imgs. with error < 18.3m	#imgs. with error > 400m
∞ / ∞	all desc.	1.4	0.5	4.7	688	13
	int. mean	1.3	0.4	5.2	674	9
900 / 2500	all desc.	1.3	0.4	5.8	671	12
	int. mean	1.5	0.5	5.5	677	11
400 / 1000	all desc.	1.5	0.5	6.4	671	12
	int. mean	1.5	0.5	6.9	671	13
[30]	all desc.	1.4	0.4	5.9	685	16
	int. mean	1.3	0.5	5.1	675	13

of feature descriptors, it can be readily combined with more memory efficient descriptors [5, 36] to further reduce the memory footprint.

The point selection algorithm from Li et al. might prefer points that form small clusters over points that are well-distributed over the model, which can lead to unstable configurations for pose estimation. The point selection scheme does not take similarity in descriptor appearance into account. As shown in Section 4.3, it thus cannot prevent a drop in registration performance when the descriptor space becomes denser. Furthermore, localization performance and efficiency depend on the set cover parameter K. An interesting open question is whether we can design a better, parameter-free point filtering algorithm that ensures a better distribution of points and impacts the descriptor space.

As shown in Section 4.2, the number of points stored in a visual word has an impact on the quality of the found correspondences. A data structure that tries to adapt the number of words to take the density of the points inside a word into account could help to improve localization performance further.

Finally, we notice that the localization methods proposed by Li et al. and Sattler et al. have both distinct strength and weaknesses, as detailed at the end of Section 3. Combining their matching directions could help to obtain a novel localization method that combines the strength of both approaches while eliminating their weaknesses, which would have a positive impact on its performance.

Acknowledgments

We gratefully acknowledge support by UMIC (DFG EXC 89) and Mobile ACcess (EFRE 280401102). We thank all participants of the Dagstuhl Seminar 11261, "Outdoor and Large-Scale Real-World Scene Analysis", for their encouraging comments and helpful and interesting discussions.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a Day. In: IEEE 12th International Conference on Computer Vision, pp. 72–79. IEEE (2009)
2. Arth, C., Wagner, D., Klopschitz, M., Irschara, A., Schmalstieg, D.: Wide Area Localization on Mobile Phones. In: 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 73–82. IEEE Comp. Society, Washington DC (2009)
3. Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., Wu, A. Y.: An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. J. ACM 45, 891–923 (1998)
4. Avrithis, Y., Kalantidis, Y., Tolias, G., Spyrou, E.: Retrieving Landmark and Non-Landmark Images from Community Photo Collections. In: Proceedings of the International Conference on Multimedia, pp. 153–161. ACM, New York (2010)
5. Bay, H., Ess, A., Tuytelaars, T. Van Gool, L.: SURF: Speeded Up Robust Features. In: Computer Vision and Image Understanding 110, 346–359 (2008)
6. Castle, R.O., Klein, G., Murray, D.W.: Video-rate Localization in Multiple Maps for Wearable Augmented Reality. In: 12th IEEE International Symposium on Wearable Computers, pp. 15–22 (2008)
7. Chen, D. M., Baatz, G., Köser, K., Tsai, S. S., Vedantham, R., Pylvänäinen, T., Roimela, K.,and Chen, X., Bach, J., Pollefeys, M., Girod, B., Grzeszczuk, R.: City-scale Landmark Identification on Mobile Devices. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 737–744. IEEE (2011)
8. Chum, O., Matas, J., Obdržálek, S.: Enhancing RANSAC by Generalized Model Optimization. In: Hong, K.-S., Zhang, Z. (eds.) Proceedings of the Asian Conference on Computer Vision. vol. 2, pp. 812–817. Asian Fed. of Comp. Vis. Societies (2004)
9. Chum, O. and Matas, J.: Optimal Randomized RANSAC. Trans. Pattern Analysis and Machine Intelligence 30, 1472–1482 (2008)
10. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D. P.: Discrete-Continuous Optimization for Large-Scale Structure from Motion. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3001–3008. IEEE (2011)
11. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. Int. J. Robotics Research 27, 647–665 (2008)
12. Eade, E., Drummond, T.: Scalable Monocular SLAM. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 469–476. IEEE Comp. Society, Washington DC (2006)
13. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. ACM 24, 381–395 (1981)
14. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building Rome on a Cloudless Day. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Proceedings of the 11th European Conference on Computer Vision: Part IV. LNCS, vol. 6314, pp. 368–381. Springer, Berlin / Heidelberg (2010)
15. Gammeter, S., Bossard, L., Quack, T., Van Gool, L.: I know what you did last summer: object-level auto-annotation of holiday snaps. In: IEEE 12th International Conference on Computer Vision, pp. 614–621. IEEE (2009)
16. Haralick, R.M., Lee, C.-N., Ottenberg, K., Nölle, M.: Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. Int. J. Comp. Vision 13, 331–356 (1994)

17. Hartley, R. I., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edition. Cambridge University Press, Cambridge (2004)
18. Havlena, M., Torii, A., Pajdla, T.: Efficient Structure from Motion by Graph Optimization. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Proceedings of the 11th European Conference on Computer Vision: Part II. LNCS, vol. 6312, pp. 100–113. Springer, Berlin / Heidelberg (2010)
19. Hays, J., Efros, A. A.: IM2GPS: estimating geographic information from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
20. Irschara, A., Zach, C., Frahm, J.-M., Bischof, H.: From Structure-from-Motion Point Clouds to Fast Location Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 2599–2606. IEEE (2009)
21. Josephson, K., Byröd, M.: Pose Estimation with Radial Distortion and Unknown Focal Length. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 2419–2426. IEEE (2009)
22. Knopp, J., Sivic, J., Pajdla, T.: Avoiding confusing features in place recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Proceedings of the 11th European Conference on Computer Vision: Part I. LNCS, vol. 6311, pp. 748–761. Springer, Berlin / Heidelberg (2010)
23. Li, Y., Snavely, N., Huttenlocher, D. P.: Location Recognition Using Prioritized Feature Matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Proceedings of the 11th European Conference on Computer Vision: Part II. LNCS, vol. 6312, pp. 791–804. Springer, Berlin / Heidelberg (2010)
24. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comp. Vision 60, 91–110 (2004)
25. Muja, M., Lowe, D. G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: International Conference on Computer Vision Theory and Application, pp. 331–340. INSTICC Press (2009)
26. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 2161–2168. IEEE (2006)
27. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8. IEEE (2007)
28. Pollefeys, M., Nister, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.-J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed Real-Time Urban 3D Reconstruction From Video. Int. J. Comp. Vision 78, 143–167 (2008)
29. Robertson, D., Cipolla, R.: An Image-Based System for Urban Navigation. In: Hoppe, A., Barman, S., Ellis, T. (eds.) The 15th British Machine Vision Conference, pp. 819–828. BMVA (2004)
30. Sattler, T., Leibe, B., Kobbelt, L.: Fast Image-Based Localization using Direct 2D-to-3D Matching. In: IEEE 13th International Conference on Computer Vision, pp. 667–674. IEEE (2011)
31. Schindler, G., Brown, M., Szeliski, R.: City-Scale Location Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–7. IEEE (2007)
32. Stephen, S., Lowe, D., Little, J.: Global Localization using Distinctive Visual Features. In: International Conference on Intelligent Robots and Systems, pp. 226–231. (2002)
33. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: Proceedings of the Ninth IEEE International Conference

on Computer Vision, vol. 2, pp. 1470–1477. IEEE Comp. Society, Washington DC (2003)
34. Snavely, N., Seitz, S. M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. In: SIGGRAPH Conference Proceedings, pp. 835–846. ACM, New York (2006)
35. Strecha, C., Pylvanainen, T., Fua, P.: Dynamic and Scalable Large Scale Image Reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 406–413. IEEE (2010)
36. Strecha, C., Bronstein, A. M., Bronstein, M. M., Fua, P.: LDAHash: Improved matching with smaller descriptors. EPFL-REPORT-152487 (2010)
37. Wendel, A., Irschara, A., Bischof, H.: Natural Landmark-based Monocular Localization for MAVs. In: IEEE International Conference on Robotics and Automation, pp. 5792–5799. IEEE (2011)
38. Weyand, T., Leibe, B.: Discovering Favorite Views of Popular Places with Iconoid Shift. In: IEEE 13th International Conference on Computer Vision, pp. 1132–1139. IEEE (2011)
39. Zamir, A. R., Shah, M.: Accurate Image Localization Based on Google Maps Street View. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Proceedings of the 11th European Conference on Computer Vision: Part IV. LNCS, vol. 6314, pp. 255–268. Springer, Berlin / Heidelberg (2010)
40. Zhang, W., Kosecka, J.: Image Based Localization in Urban Environments. In: 3rd International Symposium on 3D Data Processing, Visualization and Transmission, pp. 33–40. IEEE Comp. Society, Washington DC (2006)